IN THE CLAIMS

This is a complete and current listing of the claims, marked with status identifiers in parentheses. The following listing of claims will replace all prior versions and listings of claims in the application.

1.   (Previously Presented) A method for generating an identification value for identifying an electronic message in a Message Authentication Code (MAC) function by application of at least one first hash function with fixed compression that compresses n blocks of data into a number of blocks which is smaller than n or into one single block, the hash function being repetitively applied in a tree structure compression of the message, so that the message is being compressed in a plurality of tree-structure levels, each level receiving mi input blocks for compression, subscript i denoting a current level in the tree structure, whereby intermediate resulting numbers are produced, wherein said at least one first hash function additionally receives at least one cryptographic key as an input, and wherein different cryptographic keys are used in different levels of the tree structure, the method comprising processing an output of the tree structure compression further to obtain said identification value, characterized in that

- a residual data block is passed without compression from the current level to another, subsequent level in case n does not divide the number of input blocks m, for said current level i, and in that

- data which represent a length L of the message are concatenated to the output to obtain a concatenated output, and/or to at least one of the intermediate resulting numbers.

2. (Previously Presented) A method according to claim 1, further comprising the step of inserting a set of predefined data at a predetermined position in the message, e.g. by appending the set of predefined data to the message, so that the length of the message with the appended set of data becomes a multiple of the length of the blocks.

3. (Currently Amended) A method according to claim 1 ~~or 2~~, wherein the tree-structure compression is performed until the number of blocks is less than n.

4. (Previously Presented) A method according to claim 3, wherein said length L represents the length of the message without said appended set of data.

5. (Previously Presented) A method according to claim 4, wherein a hash function is applied to the concatenated output to obtain a compressed concatenated output, said hash function being one of:

      - the at least one first hash function; and

      - a second hash function.

6. (Currently Amended) A method according to ~~any of the preceding~~ claims 1, further comprising applying a further hash function to at least one of:

      - said output,

      - a further set of data derived from said output,

      - said concatenated output, and

      - said compressed concatenated output.

7. (Currently Amended) A method according to ~~any of the preceding~~ claims 1, further comprising applying a cryptographic function to said output or to a further set of data derived from said output.

8. (Currently Amended) A method according to claim 6 ~~or 7~~, wherein at least one of:

- said second hash function; and

- said further hash function

makes use of at least one cryptographic key.

9.   (Previously Presented) A method according to claim 8, wherein different cryptographic keys are used in one level of the tree structure.

10.   (Previously Presented) A method according to claim 8, wherein the same cryptographic key is used in a single level of the tree structure.

11.   (Currently Amended) A method according to ~~any of the preceding~~ claims 1, wherein at least one of:

- said first hash function;

- said second hash function; and

- said further hash function

is a universal hash function.

12.   (Currently Amended) A method according to ~~any of the preceding~~ claims 1, wherein at least one of:

- said at least one first hash function;

- said second hash function; and

- said further hash function

comprises at least two different hash functions.

13.   (Previously Presented) A method according to claim 12, wherein the at least two different hash functions

compress different numbers n of blocks.

14.   (Currently Amended) A method according to claim 12 ~~or 13~~, wherein at least one of the at least two different hash functions compresses a variable number n of blocks.

15.   (Currently Amended) A method according to ~~any of~~ claims 12-14, wherein the different hash functions use different cryptographic keys.

16.   (Currently Amended) A method according to ~~any of~~ claims 8-15, comprising performing a plurality of tree structure compressions of the message to obtain a plurality of results, and concatenating the plurality of results into a concatenated result.

17.   (Previously Presented) A method according to claim 16, wherein different cryptographic keys are applied in the plurality of tree-structure compressions.

18.   (Previously Presented) A method according to claim 16, wherein partly identical cryptographic keys are applied in the plurality of tree-structure compressions.

19.   (Currently Amended) A computer system comprising a memory and a processor, the processor being programmed to carry out the method of ~~any of~~ claims 1-18.

20.   (Currently Amended) A computer program product comprising means for performing the method of ~~any of~~ claims 1-18.

21.   (Currently Amended) A method for generating an identification value for identifying an electronic message in a Message Authentication Code (MAC) function by application of at least one first hash function with fixed compression that compresses n blocks of data into a number of blocks which is smaller than n or into one single block, the hash function being repetitively applied in a tree structure compression of the message, so that the message is being compressed in a plurality of tree-structure levels, each level receiving m_i input blocks for compression, subscript i denoting a current level in the tree structure, wherein said at least one first hash function additionally receives at least one cryptographic key as an input, and

wherein different cryptographic keys are used in different levels of the tree structure, the method comprising processing an output of the tree-structure compression further to obtain said identification value,

~~characterized in that~~ wherein

the method comprises determining whether or not n divides the number of input blocks mi for said current level i; and

if n does divide m_i: applying said at least one first hash function m_i/n times;

if n does not divide m_i:

- applying said at least one first hash function at most m_i/n times, whereby at least one residual data block is left unprocessed by the first hash function; and

- processing said at least one unprocessed data block by means of an auxiliary hash function which, in one single hash operation, compresses the at least one unprocessed data block into one single block, the auxiliary hash function having a compression rate, which is different from the compression rate of said first hash function.

22.   (Previously Presented) A method according to claim 21, further comprising the step of inserting a set of predefined data at a predetermined position in the message, e.g. by appending the set of predefined data to the message, so that the length of the message with the appended set of data becomes a multiple of the length of the blocks.

23.   (Currently Amended) A method according to claim 21 ~~or 22~~, wherein the tree-structure compression is performed until the number of blocks is less than n.

24.   (Previously Presented) A method according to claim 23, further comprising the step of concatenating the output with data which represent a length L of the message to obtain a

concatenated output, the length L representing the length of the message without said appended set of data.

25.   (Previously Presented) A method according to claim 24, wherein a hash function is applied to the concatenated output to obtain a compressed concatenated output, said hash function being one of:

      - the at least one first hash function; and

      - a second hash function.

26.   (Currently Amended) A method according to ~~any of~~ claims 21-~~25~~, further comprising applying a further hash function to at least one of:

      - said output,

      - a further set of data derived from said output,

      - said concatenated output, and

      - said compressed concatenated output.

27.   (Currently Amended) A method according to ~~any of~~ claims 21-~~26~~, further comprising applying a cryptographic function to said output or to a further set of data derived from said output.

28.   (Currently Amended) A method according to ~~any of~~ claims 21-~~27~~, wherein at least one of:

      - said at least one first hash function;

      - said second hash function; and

      - said further hash function

makes use of at least one cryptographic key.

29.   (Previously Presented) A method according to claim 28, wherein different

cryptographic keys are used in one

level of the tree structure.

30.   (Previously Presented) A method according to claim 28, wherein the same

cryptographic key is used in a single level of the tree structure.

31.   (Currently Amended) A method according to any of claims 21-30, wherein at least

one of:

        - said first hash function;

        - said second hash function; and

        - said further hash function

is a universal hash function.

32.   (Currently Amended) A method according to any of claims 21-31, wherein at least

one of:

        - said at least one first hash function;

        - said second hash function; and

        - said further hash function

comprises at least two different hash functions.

33.   (Previously Presented) A method according to claim 32, wherein the at least two

different hash functions compress different numbers n of blocks.

34.   (Currently Amended) A method according to claim 32 or 33, wherein at least one of

the at least two different hash functions compresses a variable number n of blocks.

35.   (Currently Amended) A method according to any of claims 32-34, wherein the

different hash functions use different cryptographic keys.

36.   (Currently Amended) A method according to ~~any of~~ claims 28-~~35~~, comprising performing a plurality of tree structure compressions of the message to obtain a plurality of results, and concatenating the plurality of results into a concatenated result.

37.   (Previously Presented) A method according to claim 36, wherein different cryptographic keys are applied in the plurality of tree-structure compressions.

38.   (Previously Presented) A method according to claim 36, wherein partly identical cryptographic keys are applied in the plurality of tree-structure compressions.

39.   (Currently Amended) A computer system comprising a memory and a processor, the processor being programmed to carry out the method of ~~any of~~ claims 21-~~38~~.

40.   (Currently Amended) A computer program product comprising means for performing the method of ~~any of~~ claims 21-~~38~~.

41.   (Previously Presented) A method for generating an identification value for identifying an electronic message, the method comprising the steps of:

        - processing at least one block of a set of data derived from the message into a resulting number by means of a delta-universal hash function ; and

        - computing a sum of the resulting number and a further block of data derived from the message to obtain a modified resulting number;

        - using the modified resulting number further to obtain said identification value.

42.   (Previously Presented) A method according to claim 41, wherein the hash function operates on a single block of data only.

43.   (Currently Amended) A method according to claim 41 ~~or 42~~, wherein the delta-universal hash function is repetitively applied in a tree-structure compression of the message, so that the message is being compressed in a plurality of tree-structure levels, each tree-

structure receiving m, input blocks for compression, the delta-universal hash function and the subsequent step of adding performing a compression of n data blocks into one single data block.

44.   (Previously Presented) A method according to claim 43, wherein a residual data block is passed without processing thereof from a current level to another subsequent level in case n does not divide the number of input blocks mi for said current level i.

45.   (Currently Amended) A method according to ~~any of~~ claims 41-44, wherein the modified resulting number is determined by the function:

$(m_1+k \bmod 2^{32}) \cdot (LSR(m_1,32)+LSR(k,32) \bmod 2^{32})+m_2 \bmod 2^{64}$,

where m_1 and m_2 denote two of said blocks of data, $LSR(x,y)$ denotes a logical-shift-right by y bits of input x, and k denotes a cryptographic key, whereby ml, m2 and k are represented as 64 bit unsigned integers.

46.   (Currently Amended) A computer system comprising a memory and a processor, the processor being programmed to carry out the method of ~~any of~~ claims 41-45.

47.   (Currently Amended) A computer program product comprising means for performing the method of ~~any of~~ claims 41-45.